Programación Python para Machine Learning

Lección 1: Preparación del entorno y carga de datos

eip ESCUELA INTERNACIONAL POSGRADOS

ÍNDICE

Lección 1: Preparación del entorno y carga de datos

1. Introducción

El crecimiento general que está experimentando Python puede convertirlo, si no lo es ya, en la plataforma dominante para Machine Learning. La principal razón para adoptar Python a la hora de implementar métodos de Machine Learning es que se trata de un lenguaje de propósito general que se puede utilizar tanto en I+D como en producción. En esta lección estudiaremos ecosistema de Python para Machine Learning. Concretamente, se abordarán los siguientes aspectos:

- Python y su creciente uso en Machine Learning.
- SciPy y la funcionalidad que proporciona con NumPy, Matplotlib y Pandas.
- Scikit-learn como módulo que proporciona todos los algoritmos de aprendizaje automático.
- Cómo configurar Python para Machine Learning y qué versiones utilizar.

Objetivos

- Conocer la potencialidad de Python y sus módulos en Machine Learning.
- Saber instalar y configurar el ecosistema Python para trabajar en Machine Learning.
- Familiarizarse con los formatos de archivo que contienen los datos en Machine Learning.
- Aprender a cargar conjuntos de datos para Machine Learning en Python.

1.1. Python

Python es un lenguaje de programación interpretado de propósito general. Por sus características, es fácil y rápido de aprender y utilizar. Tim Peters describe la filosofía de Python se recoge en el Zen de Python como se puede ver en la Ilustración 1.

¡Error! No se encuentra el origen de la referencia.

```
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>
```

Ilustración 1: El Zen de Python, por Tim Peters.

En términos generales, Python es actualmente uno de los lenguajes de programación más populares. Aparece constantemente en el top 10 de los lenguajes de programación en las preguntas en StackOverflow y ocupa el segundo puesto del ranking de lenguajes Tiobe (septiembre 2021). Se trata de un lenguaje dinámico y muy adecuado para el desarrollo interactivo, la creación rápida de prototipos y, a su vez, tiene la potencia necesaria para soportar el desarrollo de grandes aplicaciones. También es muy utilizado en Machine Learning y Ciencia de los datos debido al excelente soporte que ofrecen las librerías y módulos disponibles para ello.

Todo ello implica un aspecto a la vez simple pero muy importante: se puede realizar un proceso de investigación y desarrollo en el mismo lenguaje de programación que se utiliza para posteriores sistemas de producción, simplificando enormemente la transición del desarrollo a la producción.

1.2. SciPy, NumPy, Pandas, Matplotlib

SciPy (https://scipy.org/) es un ecosistema de software de código abierto basado en Python para las matemáticas, la ciencia y la ingeniería. Se trata de un complemento que Python necesario para desarrollar cualquier proyecto basado en técnicas de Machine Learning. En particular, el ecosistema SciPy se compone, entre otros, de los siguientes módulos relevantes para Machine Learning:

- NumPy: Una base para SciPy que le permite trabajar eficientemente con datos en arrays.
- Matplotlib: Permite crear gráficos y diagramas 2D a partir de los datos.
- Pandas: Herramientas y estructuras de datos para organizar y analizar conjuntos de datos.

Para ser eficaz en el aprendizaje automático en Python se deben instalar y familiarizarse con los módulos SciPy. Específicamente, entre las tareas que deben dominarse se incluye:

- Preparar los datos en estructuras de datos como los *arrays* de NumPy, con el fin de realizar un modelado mediante el uso de algoritmos de Machine Learning.
- Utilizar las herramientas que incluye Matplotlib, y Matplotlib contenido en otros frameworks, para crear gráficos y mostrar los datos.
- Utilizar Pandas para cargar, explorar y entender mejor los datos.

1.3. Scikit-learn

El módulo **scikit-learn** pone a disposición de la comunidad las herramientas necesarias para desarrollar y ejecutar proyectos de Machine Learning en Python. Se basa y requiere del ecosistema SciPy. Estas librerías se centran en los algoritmos de Machine Learnig para clasificación, regresión, clustering, entre otros. También proporciona los métodos para llevar a cabo tareas relacionadas, como la evaluación de modelos, el ajuste de parámetros e hiperparámetros y el preprocesamiento de los datos.

Al igual que Python y SciPy, scikit-learn es de código abierto y puede utilizarse comercialmente bajo la licencia BSD. Mediante su utilización se puede aprender sobre Machine Learning, desarrollar modelos y ponerlos en producción, todo en el mismo marco y con el mismo código.

1.4. Instalación de SciPy y Scitkit-learn

Existen múltiples maneras de instalar SciPy. Las maneras más populares son utilizar la gestión de paquetes del propio sistema operativo, utilizar la herramienta de gestión de paquetes de Python *pip* o hacer uso de Anaconda. La documentación de SciPy es excelente y cubre las instrucciones de cómo realizar el proceso sobre distintas plataformas.

Los siguientes paquetes se han de instalar como mínimo:

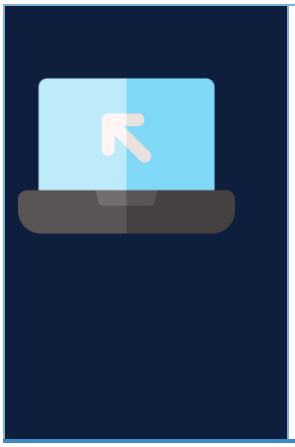
- scipy
- numpy
- matplotlib
- pandas

Una vez instalados, se debe confirmar que la instalación se ha realizado correctamente. Para ello, es necesario ejecutar el siguiente código de Python para imprimir las versiones de las bibliotecas instaladas:

```
# scipy
import scipy
print('scipy:', scipy.__version__)
# numpy
import numpy
print('numpy:', numpy.__version__)
# matplotlib
import matplotlib
print('matplotlib:', matplotlib.__version__)
# pandas
import pandas
print(' pandas:', pandas.__version__)
```

En el caso de la instalación de **scikit-learn**, el procedimiento puede ser el mismo: gestor de paquetes, herramienta *pip* o Anaconda. Igualmente, se puede confirmar que la correcta instalación de scikit-learn mediante la ejecución del siguiente código en Python:

```
# scikit-learn
import sklearn
print('sklearn:',sklearn.__version__)
```



Para más información

Si visitas el siguiente enlace conocerás en profundidad las distintas alternativas de instalación.

http://scipy.org/installation

2. CARGA DE DATOS

La primera de las tareas a completar dentro de un proyecto de Machine Learnig es poder cargar los datos. El formato más común en el que se presentan los datos en Machine Learning son los llamados archivos CSV (Comma Separated Values). Existen varias formas de cargar un archivo CSV en Python. En esta sección se estudiarán tres formas para cargar los datos CSV en Python:

- Cargar archivos CSV con la biblioteca estándar de Python.
- Cargar archivos CSV con NumPy.
- Cargar archivos CSV con Pandas.

2.1. Aspecto a considerar en la carga de archivos CSV

Hay una serie de consideraciones a tener en cuenta cuando se procede con la carga de datos contenidos en archivos CSV.

Cabecera del archivo

El archivo con los datos puede contener una primera línea a modo cabecera. En tal caso, esto significa una ayuda a la hora de asignar automáticamente nombres a cada columna de datos. Si no es así, es posible que se tenga que nombrar los atributos manualmente. En cualquier caso, se debería especificar explícitamente si el CSV tiene o no una cabecera en el momento de la carga de los datos.

Comentarios

Los comentarios en un CSV vienen indicados con una almohadilla "#" al principio de línea. Si el archivo incluye alguna línea con comentarios, dependiendo del método utilizado para cargar sus datos, es posible que se tenga que indicar si se esperan o no comentarios y el carácter que separación que los identifican.

Delimitador

El delimitador estándar que separa los valores en los campos es el carácter coma "," . En algunos casos, los archivos CSV pueden utilizar un delimitador diferente, tales como el tabulador o el espacio en blanco, en cuyo caso se deberá especificar explícitamente.

Comillas

A veces los valores de los campos pueden tener espacios. En ese caso, los valores necesitan ir entrecomillados. El carácter de las comillas por defecto es el de las dobles comillas. Se pueden utilizar otros caracteres, y puede ser que se necesite especificar el carácter de comillas utilizado en el archivo.

Para más información

Como referencia sobre el formato CSV, se puede revisar el estándar RFC 4180 titulado *Common Format and MIME Type for Comma-Separated Values (CSV) Files*

https://datatracker.ietf.org/doc/html/rfc4180

2.2. Carga de archivos CSV con las librerías de Python

Para hacer una demostración del procedimiento de carga de datos en Python con distintas librerías, se ha seleccionado el dataset *"Indian Liver Patient"*. Este conjunto de datos contiene 416 registros de pacientes con problemas de hígado y 167 registros de pacientes sin dicha patología. El conjunto de datos se recogió en el noreste de Andhra Pradesh, India. En particular, 441 de los registros pertenecen a pacientes de género masculino y 142 son de pacientes femeninos.

Se trata de un dataset utilizado para modelar un problema de clasificación. Además, representa un buen conjunto de datos de demostración porque incorpora atributos de entrada numéricos, atributos de entrada nominales y la variable de salida a predecir es binaria.

Importante

El repositorio UCI Machine Learning alberga una colección de bases y conjuntos de datos que son utilizados por la comunidad para el análisis empírico de algoritmos de Machine Learning.

https://archive.ics.uci.edu/ml/datasets.php

Python trae consigo el **módulo csv** y la función *reader()* que puede utilizarse para cargar archivos CSV. Una vez cargados, para procesarlos, se puede convertir los datos CSV en una matriz NumPy.

Igualmente, la carga de los datos CSV se puede llevar a cabo usando **NumPy** y la función *numpy.loadtxt().* Esta función asume que no hay fila de encabezado y que todos los datos tienen el mismo formato.

Por último, se pueden cargar los datos CSV utilizando **Pandas** y la función *pandas.read_csv()*. Esta función es muy flexible y es quizás el método más recomendado para cargar datos para trabajar en Machine Learning. La función devuelve un pandas.DataFrame con el que se puede comenzar directamente a trabajar en tareas de estadística descriptiva.

En el ejemplo siguiente se asume que el archivo *'Indian-Liver-Patient.csv'* está en tu directorio de trabajo actual.

```
filename = 'Indian-Liver-Patient.csv'

# Carga de datos con el módulo csv de Python
import csv
import numpy
raw_data = open(filename, 'rt')
reader = csv.reader(raw_data, delimiter=',', quoting=csv.QUOTE_NONE)
x = list(reader)
#data = numpy.array(x[]).astype('float')
#print(data.shape)

# Carga de datos CSV con NumPy
from numpy import loadtxt

raw_data = open(filename, 'rt')
data = loadtxt(raw_data, delimiter=",", dtype=str)
print(data.shape)

# Carga de datos CSV Pandas
from pandas import read_csv

col_names = ['age', 'gen', 'tbili', 'dbili', 'alkphos', 'sgpt',
'sgot', 'tp', 'alb', 'ag', 'class']
data = read_csv(filename, names=col_names)
print(data.shape)
```

Una vez vistos las distintas alternativas para la carga de datos desde un archivo CSV, se recomienda realizar el proceso utilizando Pandas, por lo que será el utilizado para todos los ejemplos siguientes de este curso.

3. PUNTOS CLAVE

En esta lección hemos aprendido:

- Valorar la verdadera potencialidad de Python y los módulos disponibles para el desarrollo de proyectos de Machine Learning.
- Instalar y configurar el ecosistema Python con los módulos necesarios para trabajar en Machine Learning.
- Descubrir el formato de archivo CSV, formato utilizado para almacenar los datos utilizados en Machine Learning.
- Cargar mediante Python conjuntos de datos correspondientes a problemas de Machine Learning. .